CLAIMS

What is claimed is:

1. A method for searching, comprising:

splitting among parallel processing blocks elements of a set of values derived form a set of ratios;

computing in parallel processing blocks a set of values derived from a set of ratios, each value of the set computed by a respective processing block;

comparing in the parallel processing blocks the respective computed value against a predetermined value accessible by the respective processing block;

selecting one of the computed value and the predetermined value for a respective processing block that is nearer to an optimum value; and

determining which of the selected values among the processing blocks is nearest to the optimum value.

2. A method according to claim 1, wherein splitting among parallel processing blocks elements of a set of values derived form a set of ratios comprises splitting among the parallel processing blocks a set of pre-computed values derived from the set of ratios, each pre-computed value of the set associated with a respective processing block.

3. A method according to claim 1, wherein splitting among parallel processing blocks elements of a set of values derived form a set of ratios comprises

computing in parallel processing blocks the set of values derived from the set of ratios, each value of the set computed by a respective processing block.

4. A method according to claim 3, wherein computing the set of values derived from the set of ratios comprises creating a ratio of an element at an index of a first buffer to an element at a corresponding index of a second buffer.

5. A method according to claim 4, wherein creating the ratio comprises creating a ratio of a square of an element of a correlation vector to an element at a corresponding index of an energy vector in a codebook search.

6. A method according to claim 4, wherein comparing the computed value to the predetermined value comprises comparing the computed ratio to a predetermined ratio.

7. A method according to claim 6, wherein comparing the computed ratio to the predetermined ratio further comprises:

generating a first product of the numerator of the computed ratio multiplied by the denominator of the predetermined ratio;

generating a second product of the numerator of the predetermined ratio multiplied by the denominator of the computed ratio; and

determining whether the first product minus the second product is greater than zero.

8. A method according to claim 7, wherein selecting one of the computed value and the predetermined value that is nearer to the optimum value comprises selecting the computed value if the first product minus the second product is greater than zero, otherwise selecting the predetermined value.

9. A method according to claim 6, wherein comparing the computed ratio to the predetermined ratio further comprises:

generating a first product of the numerator of the computed ratio multiplied by the denominator of the predetermined ratio;

generating a second product of the numerator of the predetermined ratio multiplied by the denominator of the computed ratio; and

determining whether the first product minus the second product is less than zero.

10. A method according to claim 9, wherein selecting one of the computed value and the predetermined value that is nearer to the optimum value comprises selecting the computed value if the first product minus the second product is less than zero, otherwise selecting the predetermined value.

11. A method according to claim 6, wherein comparing the ratio to the predetermined value comprises comparing the ratio to an initial-value ratio for the respective processing block.

12. A method according to claim 6, wherein comparing the ratio to the predetermined value comprises comparing the ratio to a previously computed ratio determined on a previous iteration by the respective processing block to be nearer to the optimum value than a predetermined value of the previous iteration.

13. A method according to claim 1, wherein selecting one of the computed value and the predetermined value that is nearer to the optimum value comprises selecting the greater of the computed value and the predetermined value.

14. A method according to claim 1, wherein the set of values comprises buffer elements obtained from buffers accessible by the respective processing blocks, and

wherein selecting one of the computed value and the predetermined value that is nearer to the optimum value comprises:

storing as the predetermined value in a storage medium accessible by the respective processing block one of the computed value and the predetermined value that is nearer to the optimum value; and

repeating the elements of computing, comparing, and selecting until all available buffer elements have been accessed.

15. A method according to claim 1, wherein determining which of the selected values among the processing blocks is nearest to the optimum value comprises:

if there are two selected values, repeating the elements of comparing and selecting in a processing block, with the first selected value as the predetermined value and the second selected value as the computed value; and

if there are more than two selected values, repeating in parallel processing blocks the elements of comparing and selecting, with the first selected value as the predetermined value and the second selected value as the computed value for each respective processing block.

16. An article of manufacture comprising a machine-accessible medium having content that provides instructions to cause an electronic device to:

computing in parallel processing blocks a set of values derived from a set of ratios, each value of the set computed by a respective processing block;

comparing in the parallel processing blocks the respective computed value against a predetermined value accessible by the respective processing block;

selecting one of the computed value and the predetermined value for a respective processing block that is nearer to an optimum value; and

determining which of the selected values among the processing blocks is nearest to the optimum value.

17. An article of manufacture of claim 16, wherein the content to provide instructions to cause the electronic device to compute the set of values derived from the set of ratios comprises the content to provide instructions to cause the electronic device to

create a ratio of an element of a first buffer to an element at a corresponding index of a second buffer.

18. An article of manufacture according to claim 17, wherein the content to provide instructions to cause the electronic device to create the ratio comprises the content to provide instructions to cause the electronic device to create a ratio of a square of an element of a correlation vector to an element at a corresponding index of an energy vector in a codebook search.

19. An article of manufacture according to claim 17, wherein the content to provide instructions to cause the electronic device to compare the computed value to the predetermined value comprises the content to provide instructions to cause the electronic device to compare the computed ratio to a predetermined ratio.

20. An article of manufacture according to claim 19, wherein the content to provide instructions to cause the electronic device to compare the computed ratio to the predetermined ratio further comprises the content to provide instructions to cause the electronic device to:

generate a first product of the numerator of the computed ratio multiplied by the denominator of the predetermined ratio;

generate a second product of the numerator of the predetermined ratio multiplied by the denominator of the computed ratio; and

compare the difference of the first product minus the second product to zero.

21. An article of manufacture according to claim 20, wherein the content to provide instructions to cause the electronic device to select one of the computed value and the predetermined value that is nearer to the optimum value comprises the content to provide instructions to cause the electronic device to:

if a maximum value is searched for, select the computed value if the first product minus the second product is greater than zero, otherwise selecting the predetermined value; and

if a minimum value is searched for, select the computed value if the first product minus the second product is less than zero, otherwise selecting the predetermined value.

22. An article of manufacture according to claim 19, wherein the content to provide instructions to cause the electronic device to compare the ratio to the predetermined value comprises the content to provide instructions to cause the electronic device to compare the ratio to an initial-value ratio for the respective processing block.

23. An article of manufacture according to claim 19, wherein the content to provide instructions to cause the electronic device to compare the ratio to the predetermined value comprises the content to provide instructions to cause the electronic device to compare the ratio to a previously computed ratio determined on a previous iteration by the respective processing block to be nearer to the optimum value than a predetermined value of the previous iteration.

24. A method of searching a set of ratios, comprising:

separating elements of vectors **A** and **B** into a number of different sets;

computing in parallel processing units a first product of an indexed element of vector **A** multiplied by a first member of an initial value pair;

computing in the parallel processing units a second product of an indexed element of vector **B** multiplied by a second member of the initial value pair;

setting, for each processing unit, the first member of the initial value pair to the value of the indexed element of vector **B**, and the second member of the initial value pair to the value of the indexed element of vector **A**, if the first product is greater than the second product for the processing unit;

indexing sequential elements of vectors **A** and **B** of the different sets;

repeating the above limitations until a predetermined number of elements of vectors **A** and **B** has been searched; and

determining which pair of resulting initial values among the parallel processing units provides a ratio of member one to member two that is nearest to an optimum value.

25. A method according to claim 24, wherein separating the elements into the number of different sets comprises separating the elements into a number of different sets, the number corresponding to a number of available processing units.

26. A method according to claim 24, wherein separating the elements into the number of different sets comprises separating the elements into a number of different

sets, the number determined, at least in part, by a number of separate buffer elements fit simultaneously on a data transfer bus from a memory to the processing units.

27. A method according to claim 24, wherein, for ratio maximization:

computing the first product comprises computing the multiplication of an element of the vector **A** of numerator elements by a denominator member of the initial value pair; and

computing the second product comprises computing the multiplication of an element of the vector **B** of denominator elements by a numerator member of the initial value pair.

28. A method according to claim 27, wherein vector **A** comprises a correlation vector and vector **B** comprises an energy vector.

29. A method according to claim 24, wherein, for ratio minimization:

computing the first product comprises computing the multiplication of an element of the vector **A** of denominator elements by a numerator member of the initial value pair; and

computing the second product comprises computing the multiplication of an element of the vector **B** of numerator elements by a denominator member of the initial value pair.

30. A method according to claim 24, wherein determining which pair of resulting initial values among the parallel processing units provides the ratio that is nearest to the optimum value comprises:

if there are two resulting initial value pairs, repeating the elements of computing and setting in a processing unit, with the values of one initial value pair as the indexed elements and the values of the other initial value pair as the initial value pair; and

if there are more than two resulting initial value pairs, repeating the elements of computing and setting in parallel processing units, with the values of one initial value pair as the indexed elements and the values of another initial value pair as the initial value pair for each respective processing block.

31. A apparatus comprising:

control logic to separate elements of a vector **A** and a vector **B** into a number of different sets and set a pointer to index various elements of vectors **A** and **B**, the control logic to increment the indices in response to receiving an indication from a set of parallel processing units that the parallel processing units have completed a processing function; and

a set of parallel processing units to repeatedly receive from the control logic and process elements of vectors **A** and **B** until a predetermined number of elements of vectors **A** and **B** has been searched, by:

computing a first product of an indexed element of vector **A** multiplied by a first member of an initial value pair;

computing a second product of an indexed element of vector **B** multiplied by a second member of the initial value pair;

setting, for each processing unit, the first member of the initial value pair to the value of the indexed element of vector **B**, and the second member of the initial value pair to the value of the indexed element of vector **A**, if the first product is greater than the second product for the processing unit; and

indicating to the control logic that the iteration is complete;

selection logic to determine which pair of resulting initial values among the parallel processing units provides a ratio of member one to member two that is nearest to an optimum value.

32. An apparatus according to claim 31, further comprising a memory to store vectors **A** and **B**, communicatively coupled with parallel processing units via a direct memory access (DMA) channel.

33. An apparatus according to claim 31, wherein the control logic separates the elements into the number of different sets based on the number of parallel processing units comprises the set of parallel processing units.

34. An apparatus according to claim 31, wherein the control logic separates the elements into the number of different sets based, at least in part on, a number of separate elements of the vectors fit simultaneously on a data transfer bus from a memory to the processing units.

35. An apparatus according to claim 34, wherein the data transfer bus comprises a 64-bit bus, and the elements of vectors **A** and **B** comprise 16-bit values.

36. An apparatus according to claim 31, wherein the parallel processing units search for maximization ratios, and wherein the parallel processing units each compute the first product by multiplying an element of the vector **A** of numerator elements by a denominator member of the initial value pair, and compute the second product by multiplying an element of the vector **B** of denominator elements by a numerator member of the initial value pair.

37. An apparatus according to claim 31, wherein the parallel processing units search for minimum ratios, and wherein the parallel processing units each compute the first product by multiplying an element of the vector **A** of denominator elements by a numerator member of the initial value pair, compute the second product by multiplying an element of the vector **B** of numerator elements by a denominator member of the initial value pair.

38. A method of searching a codebook, comprising:

separating elements x_k and y_k of vectors **X** and **Y** among a number N parallel processing circuits to direct elements (x_0 and y_0), (x_N and y_N), and (x_{2N} and y_{2N}) to processing circuit 0, elements (x_1 and y_1), (x_{N+1} and y_{N+1}), and (x_{2N+1} and y_{2N+1}) to

processing circuit 1, and elements (x_{N-1} and y_{N-1}), (x_{2N-1} and y_{2N-1}), and (x_{3N-1} and y_{3N-1}) to processing circuit *N*-1, where *k* represents the index of the elements of vectors **X** and **Y**;

computing in the parallel processing circuits a product $x^2_{n,N} \cdot y_{init,N}$, where $x^2_{n,N}$ represents the square of the value of the element of vector **X** at index *n* of processing circuit *N*, $y_{init,N}$ represents an initial value for vector **Y** of processing circuit *N*, and *n* represents the index of the specific separated elements to be received by processing circuit *N*;

computing in the parallel processing circuits a product $x^2_{init,N} \cdot y_{n,N}$, where $x^2_{init,N}$ represents the square of an initial value for vector **X** of processing circuit *N*, $y_{n,N}$ represents the value of the element of vector **Y** at index *n* of processing circuit *N*, and *n* represents the index of the specific separated elements to be received by processing circuit *N*;

setting the values of the pair ($x_{init,N}$,$y_{init,N}$) to the values of ($x_{n,N}$,$y_{n,N}$) for each processing circuit *N* for which the condition ($x^2_{n,N} \cdot y_{init,N}$? $x^2_{init,N} \cdot y_{n,N}$) is satisfied, where the operator ? denotes the greater than (>) operation for ratio maximization, and denotes the less than (<) operation for ratio minimization;

incrementing each index *n* for each processing circuit *N*;

repeating the above limitations until a predetermined index *k* of vectors **X** and **Y** has been reached; and

determining which of the various pairs ($x_{init,N}$,$y_{init,N}$) is nearest to an optimum value.

39. A method according to claim 38, wherein separating the elements of vectors **X** and **Y** among *N* parallel processing circuits comprises separating the elements of vector **X** and **Y** among a number of parallel processing units which corresponds to the number of elements of the vectors that can simultaneously be transmitted on a data transfer bus coupled with the processing circuits.

40. A method according to claim 38, wherein determining which of the various pairs $(x_{init,N}, y_{init,N})$ is nearest to the optimum value further comprises:

if there are more than two resulting pairs of $(x_{init,N}, y_{init,N})$ to search, repeating the elements of computing and setting in parallel processing circuits with one pair $(x_{init,N}, y_{init,N})$ as $(x_{init,N}, y_{init,N})$, and another pair $(x_{init,N}, y_{init,N})$ as $(x_{n,N}, y_{n,N})$ for each processing circuit until there are two pairs of values remaining; and

if there are two remaining pairs of values, repeating the elements of comparing and selecting in a processing circuit, with the first pair as $(x_{init,N}, y_{init,N})$ and the second pair as $(x_{n,N}, y_{n,N})$.

41. A system comprising:

a processor having:

control logic to separate elements x_k and y_k of vectors **X** and **Y** into *N* sets, where set 0 includes elements (x_0 and y_0), (x_N and y_N), and (x_{2N} and y_{2N}), set 1 includes elements (x_1 and y_1), (x_{N+1} and y_{N+1}), and (x_{2N+1} and y_{2N+1}), and set *N*-1 includes elements (x_{N-1} and y_{N-1}), (x_{2N-1} and y_{2N-1}), and (x_{3N-1} and y_{3N-1}), each set

to be processed by a corresponding separate parallel processing circuit, where *k* represents the index of the elements of vectors **X** and **Y**;

a processing core with parallel processing circuits to repeatedly compute products $(x^2_{n,N} \bullet y_{init,N})$ and $(x^2_{init,N} \bullet y_{n,N})$, where $x^2_{n,N}$ represents the square of the value of the element of vector **X** at index *n* of processing circuit *N* and $x^2_{init,N}$ represents the square of an initial value for vector **X** of processing circuit *N*, $y_{init,N}$ represents an initial value for vector **Y** of processing circuit *N* and $y_{n,N}$ represents the value of the element of vector **Y** at index *n* of processing circuit *N*, and set the values of the pair $(x_{init,N},y_{init,N})$ to the values of $(x_{n,N},y_{n,N})$ for each processing circuit *N* for which the condition $(x^2_{n,N} \bullet y_{init,N}\ ?\ x^2_{init,N} \bullet y_{n,N})$ is satisfied, until a predetermined value of *k* has been reached; and

a value selection circuit to determine which of the various pairs $(x_{init,N},y_{init,N})$ is nearest to an optimum value; and

a modulator communicatively coupled with the processor to modulate signals for transmission over a communication channel.

42. A system according to claim 41, wherein the modulator is included in a front-end transmission circuit that prepares for transmission over a power line a signal including compressed speech and the pair $(x_{init,N},y_{init,N})$ that is determined by the processor to be nearest to the optimum value.

43. A system according to claim 42, further comprising a channel coder coupled with the modulator to prepare the signal according to a protocol associated with a communication channel on the power line.

44. A system according to claim 41, wherein the processor is adapted to perform an algebraic codec search according to the Adaptive Multi-Rate (AMR) standard.